UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/790,160	03/01/2004	Subash Kalbarga	60046.0068US01	9971

53377 7590 12/23/2009

HOPE BALDAUFF HARTMAN, LLC
Michael J. Baldauff, Jr.
1720 PEACHTREE STREET, N.W
SUITE 1010
ATLANTA, GA 30309

EXAMINER	
GUPTA, MUKTESH G	

ART UNIT	PAPER NUMBER
2444	

MAIL DATE	DELIVERY MODE
12/23/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/790,160	KALBARGA, SUBASH	
	Examiner	Art Unit	
	Muktesh G. Gupta	2444	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *01 October 2009*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1 and 3-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1 and 3-20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *10/01/2009*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. **Claims 1, 3-5, 9, 12-13, 16 and 19-20** are amended.

   Claim 2, was cancelled previously.

   Claims 1 and 3-20 have been presented examined on merits and are pending in this application.

Information Disclosure Statement

2. An initialed and dated copy of the information disclosure statements (IDS) submitted on 10/01/2009 is being considered by the examiner, signed and dated copy is attached to this office action.

Continued Examination under 37 CFR 1.114

3. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 10/01/2009 has been entered.

Response to Amendment

4. Applicant's amendment filed on 10/01/2009 necessitated updating search with new ground(s) of rejection presented in this office action.

Applicant's arguments are deemed moot in view of the following new grounds of rejection as explained here below, necessitated by Applicant's substantial amendment (i.e., "*emulating a device at the computer management device, the emulated device conforming to a second communication standard; transmitting the one or more vendor specific commands from an application programming interface of the host computer to the device emulated at the computer management device over a communications link between the host computer and the computer management device, the communications link conforming to the second communication standard; determining, at the computer management device, whether the one or more vendor specific commands are commands intended for accessing data on the device emulated by the computer management device, commands for modifying configuration data associated with the computer management device, or commands for obtaining coordinates of a user input cursor on the remote computer; in response to determining that the one or more vendor specific commands are commands for modifying configuration data associated with the computer management device or commands for obtaining coordinates of a user input cursor on the remote computer, utilizing the received vendor specific commands for communicating with the computer management device; and in response to determining that the one or more vendor specific commands are commands intended for accessing data on the device emulated by the computer management device, accessing content from a mass storage device attached to the remote computer, the*

content from the mass storage device attached to the remote computer redirected from the remote computer to the computer management device) to the claims which significantly affected the scope thereof.

Applicant's arguments with respect to **Claims 1 and 3-20** have been considered but are moot in view of the new ground(s) of rejection.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. **Claims 1 and 3-20** are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent No. 6779004 to Zintel; William Michael (hereinafter "Zintel"), and in view of U.S. Patent No. 6560641 to Powderly; Terrence W. et al., (hereinafter "Powderly").

1. (Currently Amended) A method for communicating with a computer management device, the method comprising (as stated in col. 4, lines 56-62, col. 7, lines 8-60, col. 47, lines 45-63, Zintel discloses, Universal Plug and Play (UPnP) is an open network architecture that is designed to enable simple, ad hoc communication among distributed devices and services from many vendors. UPnP leverages Internet technology and can be thought of as an extension of the Web model of mobile Web

browsers talking to fixed Web servers to the world of peer-to-peer connectivity among mobile and fixed devices. User Control Point: The set of modules that enable communication with an UPnP Controlled Device. User Control Points initiate discovery and communication with Controlled Devices, and receive Events from Controlled Devices. Controlled Device: The set of modules that perform certain tasks (e.g., printing) and communicate with a User Control Point. Controlled Devices respond to discovery requests, accept incoming communications from User Control Points and may send Events to User Control Points. Controlled Devices may also include a Presentation (e.g., Web) Server. Examples of devices that could be Controlled Devices are the VCR, DVD player or recorder, heating/ventilation/air-conditioning equipment (HVAC), lighting controller, audio/video/imaging playback device, handheld computer, smart mobile phone and the PC, and the like. Nothing prevents a single device from implementing the functionality of a User Control Point and one or more Controlled Devices at the same time. Bridge is a set of modules that enables Bridged and Legacy Devices to interact with native UPnP devices. The bridge itself exposes a collection of UPnP Controlled Devices to User Control Points. The Bridge maps between native UPnP Device Control Protocols and the underlying protocols or other control methods exposed by the Bridged and Legacy Devices. Optionally, such a device could expose UPnP Controlled Devices to Legacy Devices in the manner required by the Legacy Devices. Nothing prevents a single device from implementing the functionality of a User Control Point, one or more Controlled Devices and a Bridge at the same time. UPnP Devices support automatic discovery, identification, and configuration to achieve

interoperability in the home environment, but must also operate correctly in a managed corporate network. UPnP provides a common set of interfaces for accessing devices and services, enabling the operational unification of diverse media types. Communications protocols for Universal Plug and Play are based on industry standards, especially key Internet standards such as TCP/IP, HTML, XML, HTTP, DNS, LDAP, and others):

defining, at a host computer managed by the computer management device, one or more vendor specific commands, the vendor specific commands conforming to a first communication standard, wherein the computer management device is operative to receive video output of the host computer and transmit the video output to a remote computer and further operative to receive user input received at and transmitted from the remote computer and provide the user input to the host computer (as stated in col. 2, lines 29-57, col. 6, lines 12-16, col. 11, lines 46-52, col. 51, lines 4-26, col. 21, lines 5-20, col. 44, lines 30-54, col. 51, lines 56-65, Zintel discloses, peripheral devices connected with a host via host/peripheral connectivity are exposed in a device control model as peer devices having peer networking connectivity. A peer networking-to-host/peripheral connectivity adapter, which may be implemented as a set of software modules running on a host, operates to convert between a device control protocol with peer networking connectivity and a host/peripheral connectivity protocol (or protocols) for a set of host-connected peripheral devices. The adapter, in effect, operates virtually as a set of controlled devices in the device control protocol, which respond to communication in the device control protocol from other peer devices that are

networked with the host. Alternatively, the adapter also may operate for peripheral devices that provide a user interface as a user control point that converts communications from the devices in the respective host/peripheral protocol into the device control protocol with peer networking connectivity to control other peer networking connectivity devices. UPnP leverages formal protocol contracts to enable peer-to-peer interoperation. Protocols contracts enable real-world multiple-vendor interoperation. Control Server: The module that runs in a Control Server is the module that runs in a Controlled Device or Bridge that responds to Commands invoked on a Service by a User Control Point. Commands are encoded and sent using the Service Control Protocol (SCP) specified in the Service Definition. This service consists of a TCP/HTTP server that passes control to the native control logic of a Service, updates the Service State Table (SST)and generates an event if the SST changes. With reference now to FIG. 41, an instance of the UPnP bridge 120 (FIG. 2) is automatically installed and configured for a peripheral device by operating system software of a host computing platform in accordance with a process 1100 upon attaching or connecting the peripheral to the host. In an implementation illustrated herein, this UPnP Bridge self-install and configure process 1100 is implemented as an enhancement of the plug-and-play process of the Microsoft Windows operating system running on a personal computer. The host computer's operating system detects at action 1102 as part of it boot-up routine that any new peripheral devices have been attached to a peripheral device bus (e.g., the ISA, EISA, PCI, USB, serial or parallel ports, etc.) of the host computer. Alternatively, this detecting of new peripheral devices can be done at any

time during normal operation of the host computer so as to detect "hot-plugging" peripheral devices that can be attached to the host computer while running. Upon detecting a new peripheral device has been added, the host computer's operating system proceeds to identify and install appropriate device driver software (vendor specific) for the host operating system to interact and communicate with and/or control the peripheral device as indicated at action 1103. User Control Points 104 are not required to have any prior knowledge of the SCPs 402 required to control the Services on the various devices. Therefore, a Controlled Device or Bridge must be able to describe to a User Control Point the protocols required to control its Services, such that the User Control Point will be able to implement these protocols dynamically. This requires a standard way of declaring Service Control Protocols in a concise and unambiguous fashion. UPnP introduces a technique for declaring Service Control Protocols using a series of XML documents. A Rehydrator 410 is a module that exposes a suitable API to applications and either invokes Commands on a Service or queries the state of that Service, or receives and responds to events. The primary job of the Rehydrator is to map between API calls and the Service Control Protocol sequence that invokes the Command. A user may enter commands and information into the computer 820 through a keyboard 840 and pointing device, such as a mouse 842. The computer 820 operates in a networked environment using logical connections to one or more remote computers, such as a remote computer 849. The remote computer 849 may be a server, a router, a peer device or other common network node, and typically includes many or all of the elements described relative to the computer 820, although only a

memory storage device 850 has been illustrated in FIG. 21. The logical connections depicted in FIG. 21 include a local area network (LAN) 851 and a wide area network (WAN) 852. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the Internet. Although illustrated with reference to the Microsoft Windows operating system and a personal computer as the host, this self-install and configuring of a peer networking-to-host/peripheral adapter also can be implemented on various other host/peripheral computing platforms (such as in a video game consoles, television set-top boxes, audio-video consoles, etc., which are structured as host computers which are extensible by connecting peripheral devices) so as to automatically provide peer networking upon attaching a new peripheral to such computing platforms);

emulating a device at the computer management device, the emulated device conforming to a second communication standard; transmitting the one or more vendor specific commands from an application programming interface of the host computer to the device emulated at the computer management device over a communications link between the host computer and the computer management device, the communications link conforming to the second communication standard (as stated in col. 14, lines 35-49, col. 16, lines 7-21, col. 13, lines 56-67, col. 14, lines 1-34, Zintel discloses, The UPnP Device Model 200 shown in FIG. 3 is the model of a UPnP Controlled Device or Bridge that is emulating native Controlled Devices. The Device Model includes the addressing scheme, eventing scheme, Description Document schema, Devices and Services schema and hierarchy, and the functional description of modules. The UPnP

Device Model extends beyond simple API or a command and control protocol definitions to enable multiple User Control Points to have a consistent view of Controlled Devices. This requires that the state of running services be formally modeled and that all state changes be visible to User Control Points. Central to the distributed UPnP architecture is the rule that Controlled Devices are the ultimate authority for the state of Services running on them. According to the device model 200 shown in FIG. 3, an UPnP Device 202-205 (e.g., multiple function devices 102-103 of FIG. 1 and bridged devices 122-123 of FIG. 2) is a logical container of one or more Services 210-217. Generally a Device represents a physical entity such as a VCR. Typical Services in the VCR Device example might be "TRANSPORT", "TUNER", "TIMER" and "CLOCK". While Devices are often physical entities, a PC emulating the traditional functions of a VCR could also be modeled in the same way as the stand-alone VCR. Devices can contain other Devices. An example would be a TV/VCR 250 (FIG. 4) packaged into a single physical unit. A Device (e.g., devices 202-203) may also be a logical container of other Devices. The top-most Device in a hierarchy of nested Devices 203-205 is called the Root Device 202. A Device with no nested Devices is always a Root Device. User Control Points typically have user interface that is used to access one or more Controlled Devices on the network. Controlled Devices typically only have local user interfaces. Bridges 120 (FIG. 2) exposes devices that do not expose native UPnP protocols as native UPnP Controlled Devices. The following table lists the modules in the User Control Points 104-105 and Controlled Devices 106-107, along with their functions. (1)User Control Point: (a) Function: Execute Applications. (b) Module: Application Execution Environment. (a)

Function: Invoke Commands on a controlled Device by sending Service Control Protocols in response to local API calls. (b) Module: Rehydrator (2) Controlled Device: (a) Function: Accept incoming commands in SCP's and execute them (b) Module: Control server plus native control logic. The definition of a structured unit of data called a Service Control Protocol Declaration (SCPD). SCPD is used to advertise the layout (schema) of the SST and Command Set of the Service to a User Control Point or Bridge. The SCPD enables the User Control Point to invoke Commands (through the Rehydrator) on the Controlled Device without any prior or persistent knowledge of the capabilities of the device. The SCPD is uploaded from the Controlling Device as part of the Description Document. Generation of the SCPD for a Service based on its SST definition and Command Set definition);

receiving the one or more vendor specific commands at the computer management device (as stated in col. 15, lines 60-67, col. 20, lines 64-67, col. 21, lines 1-20, Zintel discloses, The definition of a network protocol used to invoke Commands against the SST associated with a Service and to return results. The SCP can be generated from the SCPD. The Rehydrator's job is to convert SCPDs into SCPs. The reason for a formal SCP specification is to enable the implementation of the Control Server itself and to enable simple peer-to-peer device interoperation using only published protocols. With reference now to FIG. 7, all (UPnP) Controlled Devices 106-107 (FIG. 1) or Bridges 120 (FIG. 2) expose one or more Services 210-217 (FIG. 3) that can be controlled remotely. Controlling such Services involves a message exchange between a User Control Point 104 and the device 106. This message exchange

happens according to a specific Service Control Protocol (SCP) 402, which specifies the content and sequence of the messages exchanged. User Control Points 104 are not required to have any prior knowledge of the SCPs 402 required to control the Services on the various devices. Therefore, a Controlled Device or Bridge must be able to describe to a User Control Point the protocols required to control its Services, such that the User Control Point will be able to implement these protocols dynamically. This requires a standard way of declaring Service Control Protocols in a concise and unambiguous fashion. UPnP introduces a technique for declaring Service Control Protocols using a series of XML documents. A Rehydrator 410 is a module that exposes a suitable API to applications and either invokes Commands on a Service or queries the state of that Service, or receives and responds to events. The primary job of the Rehydrator is to map between API calls and the Service Control Protocol sequence that invokes the Command);

determining, at the computer management device, whether the one or more vendor specific commands are commands intended for accessing data on the device emulated by the computer management device, commands for modifying configuration data associated with the computer management device, or commands for obtaining coordinates of a user input cursor on the remote computer (as stated in col. 21, lines 35-53, Zintel discloses, More generally with reference to FIG. 8, the Rehydrator 410 operates as a universal adapter to provide a programmatic interface to any service-specific protocol of a remote computing device. The Rehydrator 410 simply obtains a data description or declaration of the methods, properties and events of the remote

service, as well as a definition of the protocol of network data messages through which the Rehydrator invokes the methods, queries or sets the properties, and receives event notifications. In UPnP, this data description takes the form of the Description Document 226, which contains a Contract 412. The Contract defines network data packets 413 (e.g., XML data), request/response patterns, and protocol (e.g., GENA, HTTP, SSDP) via which the packets are exchanged. This information is sufficient for the Rehydrator to exchange the appropriate network data packets to interact with the Controlled. Device Service, including to invoke commands, query and set properties, and receive and respond to events);

in response to determining that the one or more vendor specific commands are commands for modifying configuration data associated with the computer management device or commands for obtaining coordinates of a user input cursor on the remote computer, utilizing the received vendor specific commands for communicating with the computer management device; and in response to determining that the one or more vendor specific commands are commands intended for accessing data on the device emulated by the computer management device, accessing content from a mass storage device attached to the remote computer, the content from the mass storage device attached to the remote computer redirected from the remote computer to the computer management device (as stated in col. 14, lines 51-67, col. 15, lines 1-17, Zintel discloses, The fundamental controllable entity in UPnP is a Service 210-217. Every running instance of a Service includes: A Service State Table (SST) 230, which represents the current state of the Service. The SST 230 can be used to represent the

operational mode of device or to act as an information source or sink for structured data or simple files. The SST of a VCR 254 (FIG. 4) could represent the current transport mode, tuner channel selection, input and output switch selections, audio and video decoding format and current timer program. The SST of clock 251 (FIG. 4) would likely represent the current time. The SST of an image rendering device could implement a video frame-buffer that can accept raw pixel information or formatted JPG files. The SST of an audio or video playback device could implement a transfer buffer or queue of material to be played. The SST of PDA could implement a collection of formatted data that has changed and needed to be synchronized with another device, in addition to a transfer buffer for accepting incoming formatted data. The logical structure of a SST published in the Service Definition, but the actual storage format of an instance of a SST is entirely up the device. The only interaction with a SST is through a formal application level network protocol. A Control Server 232, which accepts incoming Commands expressed in the Service's Service Control Protocol (SCP). The Control Server passes the command to the Service's native command processing logic and waits for command completion. When the command is completed successfully, the SST is updated, an event is generated, and a successful response is returned to the User Control Point).

3. (Currently Amended) The method of Claim 1, wherein utilizing the received vendor specific commands for communicating with the computer management device in response to determining that the one or more vendor specific commands are

commands for modifying configuration data associated with the computer management device or commands for obtaining coordinates of a user input cursor on the remote computer comprises utilizing data contained in the received vendor specific commands to configure the computer management device (as stated in col. 6, lines 31-67, col. 17, lines 49-67, col. 18, lines 1-2, Zintel discloses, Since UPnP enables the browser to be extended to control devices, and because UPnP devices are controlled with explicit protocols, the browser must somehow learn how to talk to UPnP devices. This learning process is driven entirely from the device itself and is accomplishing entirely by uploading an XML document that describes the capabilities of the device. The architectural component that enables device-driven auto-configuration is called the Rehydrator. The job of the Rehydrator is to convert between APIs and protocols. An SST can be used to represent to current operational mode of device, act as an information source or sink and/or simply be a repository for commands. The SST of a VCR Service could represent the current transport mode, tuner channel selection, input and output switch selections, audio and video decoding format and current timer program. Alternatively, the VCR 254 could be represented as a Transport Service 260, Tuner Service, I/O Switch Service, A/V Decoding Configuration Service and Programmable Timer Service 261. The SST of a clock 251 would likely represent the current time. Additionally an alarm clock could include Service Variables to configure the clock. The SST of an image rendering device could implement a video frame-buffer that can accept raw pixel information or formatted JPG files. The SST of an audio or video playback device could implement a transfer buffer or queue of material to be

played. The SST of PDA could implement a collection of formatted data that has changed and needed to be synchronized with another device, in addition to a transfer buffer for accepting incoming formatted data).

4. (Currently Amended) The method of Claim 3, wherein utilizing data contained in the received vendor specific commands to configure the computer management device comprises setting a network address of the management device based upon contents of the received vendor specific commands (as stated in col. 14, lines 35-49, Zintel discloses, The UPnP Device Model 200 shown in FIG. 3 is the model of a UPnP Controlled Device or Bridge that is emulating native Controlled Devices. The Device Model includes the addressing scheme, eventing scheme, Description Document schema, Devices and Services schema and hierarchy, and the functional description of modules. The UPnP Device Model extends beyond simple API or a command and control protocol definitions to enable multiple User Control Points to have a consistent view of Controlled Devices. This requires that the state of running services be formally modeled and that all state changes be visible to User Control Points. Central to the distributed UPnP architecture is the rule that Controlled Devices are the ultimate authority for the state of Services running on them).

5. (Currently Amended) The method of Claim 1, wherein utilizing the received vendor specific commands for communicating with the computer management device in response to determining that the one or more vendor specific commands are

commands for modifying configuration data associated with the computer management device or commands for obtaining coordinates of a user input cursor on the remote computer comprises: (as stated in col. 13, lines 52-55, col. 15, lines 10-16, col. 28, lines 54-67, col. 29, lines 1-10, col. 28, lines 54-67, Zintel discloses, Controlled Devices 106-107 are responsible for storing and updating the state of Services. User Control Points are required to synchronize to the state on Controlled Devices and to share state directly among themselves. A Control Server 232, which accepts incoming Commands expressed in the Service's Service Control Protocol (SCP). The Control Server passes the command to the Service's native command processing logic and waits for command completion. When the command is completed successfully, the SST is updated, an event is generated, and a successful response is returned to the User Control Point. With reference to FIG. 19, the UPnP architecture 200 (FIG. 3) requires that clients of the UPnP API be enabled to receive notifications reliably from UPnP services 210-217 as their states change. Since state changes are relatively common, the eventing subsystem's efficiency and performance is a major consideration in this design. FIG. 19 and the following discussion describe the Basic UPnP Eventing Architecture 600, which encompasses both the controlled device (CD) 106 and user control point (UCP) 104 sides of the eventing service. It also includes the support APIs for both a low-level service interaction and a higher level COM-based wrapper of those APIs. The latter enables automation controllers like Visual Basic and JScript 602 to receive event notifications. Property change events are defined as any change in the value of a row of the Device State Table (DST) 230 (FIG. 3) for a service 210-217. This change will be

reflected as a property change notification. For example, if a "VCR" device has a "VCR Transport" service, one row in that service's DST may be TapeState and the value could be TapePresent. If the tape is ejected, the new value would be TapeAbsent. This state change would be reflected as a notification sent to all subscribers);

determining coordinates of a user input cursor on the remote computer; and returning the coordinates to the host computer in response to the received vendor specific commands (as stated in col. 17, lines 62-67, col. 13, lines 56-60, col. 30, lines 54-67, col. 31, lines 1-13, Zintel discloses, The SST of an image rendering device could implement a video frame-buffer that can accept raw pixel information or formatted JPG files. The SST of an audio or video playback device could implement a transfer buffer or queue of material to be played. The SST of PDA could implement a collection of formatted data that has changed and needed to be synchronized with another device, in addition to a transfer buffer for accepting incoming formatted data. User Control Points typically have user interface that is used to access one or more Controlled Devices on the network. Controlled Devices typically only have local user interfaces. The Service object 612 includes a static member function called EventNotifyCallback( ) which is invoked for each notification sent by the UPnP service. The callback is passed the entire HTTP message contents in a structure which is a parameter to the function. The prototype looks like this: The:ssdpType parameter should always be SSDP_PROPCHANGE. The pssdpMsg parameter contains the relevant information about the event. The key piece of information is the body of the XML message. The body contains information about what property changed, what its new value is and what

type it is, among other information. The pContext parameter will always be the pointer of the Service object. This allows the code to call a method to fire the event to the UCP. The callback will parse the XML body using the XML DOM services. Property changes are iterated and the local DST is updated to reflect these changes. After this processing is done, an event notification may be fired for each property that was changed to the owner of the subscription if one exists. Depending on what environment the owner is written in (C++ or script, etc . . .), a different mechanism for firing the event may be employed).

6. (Previously Presented) The method of Claim 1, wherein the first communication standard comprises the SCSI standard, the second communication standard comprises the USB standard, and wherein the emulated device comprises a USB mass storage device (as stated in col. 1, lines 65-67, col. 2, lines 1-4, col. 50, lines 37-67, col. 31, lines 1-13, Zintel discloses, The prevalent model for device connectivity, has been that of host-peripheral connectivity, typified by the personal computer and its many peripheral devices (e.g., data storage drives, user input devices, displays, printers, scanners, etc.) connected via various buses (e.g., PCI, VESA, AGP, Microchannel, ISA, EISA, USB), ports (e.g., serial, parallel), and connectors (e.g., PS/2 connector). FIG. 27 illustrates a pervasive computing environment 1000, such as may be installed in a home, office or public place, which includes a large number of embedded computing devices, such as the illustrated device 900 (FIG. 22). The pervasive computing environment 1000 includes personal computers 1002, 1004 (e.g.,

of the type shown in FIG. 21) connected via a local area network (LAN) 1006. The PC 1002 is connected via a universal serial bus 1016 to a telephone modem 1010, XDSL interface 1011 or a cable modem 1012, which in turn provide a connection with the computer network, e.g., the Internet. Various embedded computing devices also connect to the computer network via various network connections to the PCs 1002, 1004. These include an audio device 1014 (e.g., speakers, radio tuner, microphone), and printer 1015 which connect to the PC 1004 through a USB 1017. Also, a digital camera 1020, a handheld PC (H/PC) 1021 and another personal computing device 1022 connect via an infrared port (IRDA) 1024, which also attaches to the PC 1004 through the USB 1017. Devices, such as a portable telephone 1050 and remote control 1051, have a radio frequency network connection with the PC 1004. With their various inter-networked connections, the embedded computing devices are "visible" and accessible from a client device 950 (FIG. 27) also connected to the computer network. Although illustrated with reference to the Microsoft Windows operating system and a personal computer as the host, this self-install and configuring of a peer networking-to-host/peripheral adapter also can be implemented on various other host/peripheral computing platforms. Having described and illustrated the principles of our invention with reference to an illustrated embodiment, it will be recognized that the illustrated embodiment can be modified in arrangement and detail without departing from such principles. It should be understood that the programs, processes, or methods described herein are not related or limited to any particular type of computer apparatus, unless indicated otherwise. Various types of general purpose or specialized computer

apparatus may be used with or perform operations in accordance with the teachings described herein. Elements of the illustrated embodiment shown in software may be implemented in hardware and vice versa.

Zintel discloses connecting and emulating various devices and peripheral (such as mass storage device)to host computer through various communication buses (e.g., PCI, VESA, AGP, Microchannel, ISA, EISA, USB), ports (e.g., serial, parallel), and connectors (e.g., PS/2 connector).

Zintel does not explicitly disclose connecting SCSI peripheral devices through SCSI communication interface (communication standard) or adapter. However does suggest various computing platforms implementation for communicating between peer networking-to-host/peripheral adapter can be implemented on various other host/peripheral computing platforms such as peripheral device interface adapter comprising a Small Computer Systems Interface (SCSI) controller. Or adapter for host-peripheral connectivity, typified by the personal computer and its many peripheral devices (e.g., SCSI data storage drives).

In related networking art, Powderly as stated in col. 2, lines 36-41, col. 2, lines 64-67, col. 3, line 1, col. 18, lines 66-67, col. 19, lines 1-35 also in particular discloses system, method, and adapter card for providing emulation of console of a host computer system from another computer system remotely located on a network, including in particular, remote control of a peripheral device, such as a data storage device, connected to the host computer system over a second communication channel such SCSI communication standard. Server program establishes communications with the

communications client program on the adapter card, and, thereafter, upon receipt of requests from the communications client program, invokes functions of the host computer system BIOS to control the peripheral device. The option available to an Administrator-level user is the Disk Configuration option which supports the peripheral device control functionality of the present invention--in this case, control of a disk drive connected to the peripheral device interface controller 48 (e.g., disk 15a or 15b of FIG. 1). Because this option requires that the modified SCSI BIOS extension of the present invention, the user must first access HTML page 110 via the Admin option of page 104 in order to choose to have the modified SCSI BIOS extension loaded and the host computer system then reset. The Select Disk option allows the user to select a storage medium (e.g., disk unit) attached to the SCSI controller 48. As described above, selecting this option will cause the communications client RPM on the adapter card to send a request to the server program 134 on the host computer system to call the appropriate INT 13 h function to select a particular disk drive.

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Zintel's emulating various devices and peripheral (such as mass storage device) to host computer through various communication buses (e.g., PCI, VESA, AGP, Microchannel, ISA, EISA, USB), ports (e.g., serial, parallel), and connectors (e.g., PS/2 connector), to incorporate Powderly's Adapter card, also a PCI card with Peripheral Device Interface Controller. The adapter card further comprises a peripheral device interface controller to which peripheral devices can be connected and through which the host computer system can access the peripheral devices. In another

embodiment, the peripheral device interface controller 48 comprises a Small Computer Systems Interface (SCSI) controller. Zintel discloses host computer's operating system identifies and install an appropriate device driver software (vendor specific) for the host operating system to interact with and/or control the peripheral device, and suggest configuring of a peer networking-to-host/peripheral adapter can be implemented on various other host/peripheral computing platforms such as SCSI communication standards. Powderly also disclose SCSI controllers to which SCSI peripheral devices can be connected for remote emulation of host systems and its attached peripherals control in absence of Host operating system through first and second communication standard.

Therefore, it would have been obvious to combine these references of Zintel and Powderly disclosure in light of guaranteed communication control for various types of peripheral device with different device interfaces or adapter.

7. *(Original) A computer-readable medium having computer executable instructions stored thereon which, when executed by a computer, cause the computer to perform the method of claim 1* (as stated in col. 43, lines 48-53, col. 44, lines 8-29, col. 6, lines 14-17, col. 7, lines 4-7, Zintel discloses, FIG. 21 and the following discussion are intended to provide a brief, general description of a suitable computer which may be used in the above described UPnP device control model. This conventional computer 820 (such as personal computers, laptops, palmtops or handheld-PCs, set-tops, servers, mainframes, and other variety computers). The computer 820 further includes a

hard disk drive 827, a magnetic disk drive 828, e.g., to read from or write to a removable disk 829, and an optical disk drive 830, e.g., for reading a CD-ROM disk 831 or to read from or write to other optical media. The drives and their associated computer-readable media provide nonvolatile storage of data, data structures, computer-executable instructions, etc. for the computer 820. Although the description of computer-readable media above refers to a hard disk, a removable magnetic disk and a CD, it should be appreciated by those skilled in the art that other types of media which are readable by a computer. A number of program modules may be stored in the drives and RAM 825, including an operating system 835, one or more application programs 836, other program modules 837, and program data 838. UPnP leverages formal protocol contracts to enable peer-to-peer interoperation. Protocols contracts enable real-world multiple-vendor interoperation and various modules component of a device, software program, or system that implements some "functionality", which can be embodied as software, hardware, firmware, electronic circuitry, or etc).

8. (Original) A computer-controlled apparatus capable of performing the method of claim 1 (as stated in col. 7, lines 8-29, Zintel discloses, User Control Points are set of modules that enable communication with a UPnP Controlled Device. User Control Points initiate discovery and communication with Controlled Devices, and receive Events from Controlled Devices. User Control Points are typically implemented on devices that have a user interface. This user interface is used to interact with Controlled Devices over the network. The modules minimally include a Discovery Client, a

Description Client, a Rehydrator, an Event Subscription Client and an Event Sink. User Control Points may also include Visual Navigation, a Web browser and an application execution environment. User Control Points can add value to the network by aggregating the control of multiple Controlled Devices (the universal remote) or they can implement a function as simple as initiating the transfer of data to or from a Controlled Device. Examples of devices that could be User Control Points are the personal computer (PC), digital television (DTV), set-top box (STB), handheld computer and smart mobile phone, and the like. Nothing prevents a single device from implementing the functionality of a User Control Point and one or more Controlled Devices at the same time).

9. (Currently Amended) A method for communicating with a computer management device, the method comprising:

emulating a mass storage device at the computer management device, the mass storage device made available on a communication link between the computer management device and a host computer managed by the computer management device, the communication link conforming to a first communication standard, wherein the computer management device is operative to receive video output of the host computer and transmit the video output to a remote computer and further operative to receive user input received at and transmitted from the remote computer and provide the user input to the host computer (as stated in col. 8, lines 28-39, col. 14, lines 35-49, col. 45, lines 22-44, Zintel discloses, In the context of the Device Model, a container for

Services. A Device generally models a physical entity such as a VCR, but can also represent a logical entity. A PC emulating the traditional functions of a VCR would be an example of a logical device. Devices can contain other Devices. An example would be a TV/VCR packaged into a single physical unit. UPnP enables the association of user interface (display icon and root Web page) with every Device, including Root Device. The UPnP Device Model 200 shown in FIG. 3 is the model of a UPnP Controlled Device or Bridge that is emulating native Controlled Devices. The Device Model includes the addressing scheme, eventing scheme, Description Document schema, Devices and Services schema and hierarchy, and the functional description of modules. The UPnP Device Model extends beyond simple API or a command and control protocol definitions to enable multiple User Control Points to have a consistent view of Controlled Devices. This requires that the state of running services be formally modeled and that all state changes be visible to User Control Points. FIGS. 22 and 23 are intended to provide a brief, general description of a suitable embedded computing device 900 which may be used in the illustrated implementation of the invention. The embedded computing device 900 can be any variety of device incorporating electronics to control operational functions (operational circuitry 906), and in which computing and networking capabilities are embedded. For example, devices in which computing and networking functions can be embedded include communications devices (e.g., telephones, cell phones, audio and video conferencing systems, 2-way radios, etc.), office equipment (printers, fax machines, copiers, dictation, etc.), audio-video equipment (audio and video recorders and players, including televisions, radio receivers, compact disk (CD), digital video disk

(DVD), camcorders, etc.), entertainment devices (set-top boxes, game consoles, etc). (Examiner considers Compact Disks and Digital Video Disks to be Mass storage devices and with the teachings of Zintel they can be emulated as Mass Storage Devices);

receiving at the computer management device, from an application programming interface of the host computer, one or more vendor specific commands directed toward the mass storage device, the vendor specific commands conforming to a second communication standard and transmitted to the computer management device over the communication link conforming to the first standard;

determining, at the computer management device, whether the received vendor specific commands are commands intended for accessing data on the mass storage device emulated by the computer management device, commands for modifying configuration data associated with the computer management device, or commands for obtaining coordinates of a user input cursor on the remote computer;

in response to determining that the one or more vendor specific commands are commands for modifying configuration data associated with the computer management device or commands for obtaining coordinates of a user input cursor on the remote computer, utilizing the received vendor specific commands for communicating with the computer,management device;

and in response to determining that the one or more vendor specific commands are commands intended for accessing data on the mass storage device emulated by the computer management device, accessing content from a mass storage device

attached to the remote computer, the content from the mass storage device attached to the remote computer redirected from the remote computer to the computer management device (As for the rest of the features of Claim 9 which are similar to Claim 1, Examiner uses the same rational as Claim 1 to reject Claim 9).

10. (Original) The method of Claim 9, wherein the first communication standard comprises the USB standard and wherein the second communication standard comprises the SCSI standard (As for the rest of the features of Claim 10 which are similar to Claim 6, Examiner uses the same rational as Claim 6 to reject Claim 10).

11. (Previously Presented) The method of Claim 9, wherein the emulated mass storage device comprises an emulated CD-ROM device on a USB communication link (as stated in col. 8, lines 28-39, col. 14, lines 35-49, col. 45, lines 22-44, Zintel discloses, FIGS. 22 and 23 are intended to provide a brief, general description of a suitable embedded computing device 900 which may be used in the illustrated implementation of the invention. The embedded computing device 900 can be any variety of device incorporating electronics to control operational functions (operational circuitry 906), and in which computing and networking capabilities are embedded. For example, devices in which computing and networking functions can be embedded include communications devices (e.g., telephones, cell phones, audio and video conferencing systems, 2-way radios, etc.), office equipment (printers, fax machines, copiers, dictation, etc.), audio-video equipment (audio and video recorders and players,

including televisions, radio receivers, compact disk (CD), digital video disk (DVD), camcorders, etc.), entertainment devices (set-top boxes, game consoles, etc). (Examiner considers Compact Disks and Digital Video Disks to be Mass storage devices and with the teachings of Zintel they can be emulated as Mass Storage Devices).

12. (Currently Amended) The method of Claim 9, wherein utilizing the received vendor specific commands for communicating with the computer management device in response to determining that the one or more vendor specific commands are commands for modifying configuration data associated with the computer management device or commands for obtaining coordinates of a user input cursor on the remote computer comprises utilizing the vendor specific commands to configure the computer management device (As for the rest of the features of Claim 12 which are similar to Claim 3, Examiner uses the same rational as Claim 3 to reject Claim 12).

13. (Currently Amended) The method of Claim 9, further comprising in response to determining that the one or more vendor specific commands are commands for obtaining coordinates of a user input cursor on the remote computer:

determining coordinates of a user input cursor on the remote computer;

and returning the coordinates to the host computer in response to the received vendor specific commands (As for the rest of the features of Claim 13 which are similar to Claim 5, Examiner uses the same rational as Claim 5 to reject Claim 13).

14. (Original) A computer-readable medium having computer executable instructions stored thereon which, when executed by a computer, cause the computer to perform the method of claim 9 (As for the rest of the features of Claim 14 which are similar to Claim 7, Examiner uses the same rational as Claim 7 to reject Claim 14).

15. (Original) A computer-controlled apparatus capable of performing the method of claim 9 (As for the rest of the features of Claim 15 which are similar to Claim 8, Examiner uses the same rational as Claim 8 to reject Claim 15).

16. (Currently Amended) A system for managing a host computer, the system comprising:

the host computer supporting a communication link that conforms to a first communication standard and including an application programming interface, the application programming interface of the host computer operative to transmit one or more vendor specific commands that conform to a second communication standard over the communication link;

and a management device for managing the host computer, the management device connected to the host computer via the communication link, the management device operative to:

receive video output of the host computer and transmit the video output to a remote computer, receive user input received at and transmitted from the remote

computer and provide the user input to the host computer, emulate a mass storage device on the communication link, receive the vendor specific commands from the application programming interface of the host computer directed toward the mass storage device, determine whether the received vendor specific commands are commands intended for accessing data on the mass storage device emulated by the management device, commands for modifying configuration data associated with the management device, or commands for obtaining coordinates of a user input cursor on the remote computer, utilize the received vendor specific commands for communicating with the management device in response to determining that the one or more vendor specific commands are commands for modifying configuration data associated with the management device or commands for obtaining coordinates of a user input cursor on the remote computer, and access content from a mass storage device attached to the remote computer in response to determining that the one or more vendor specific commands are commands intended for accessing data on the mass storage device emulated by the management device, the content from the mass storage device attached to the remote computer redirected from the remote computer to the management device (As for the rest of the features of Claim 16 which are similar to Claim 1, Examiner uses the same rational as Claim 1 to reject Claim 16).

17. (Original) The system of Claim 16, wherein the first communication standard comprises the USB standard and wherein the second communication standard

comprises the SCSI standard (As for the rest of the features of Claim 17 which are similar to Claim 6, Examiner uses the same rational as Claim 6 to reject Claim 17).

18. (Previously Presented) The system of Claim 16, wherein the emulated mass storage device comprises an emulated CD-ROM device on a USB communication link (As for the rest of the features of Claim 18 which are similar to Claim 11, Examiner uses the same rational as Claim 11 to reject Claim 18).

19. (Currently Amended) The system of Claim 16, wherein the management device is further operative to utilize the received vendor specific commands to configure the management device in response to determining that the one or more vendor specific commands are commands for modifying configuration data associated with the management device (As for the rest of the features of Claim 19 which are similar to Claim 3, Examiner uses the same rational as Claim 3 to reject Claim 19).

20. (Currently Amended) The system of Claim 16, wherein in response to determining that the one or more vendor specific commands are commands for obtaining coordinates of a user input cursor on the remote computer, the management device is further operative to:

determine coordinates of a user input cursor on the remote computer; and return the coordinates to the computer in response to the received vendor specific

commands (As for the rest of the features of Claim 20 which are similar to Claim 5, Examiner uses the same rational as Claim 5 to reject Claim 20).

Remarks

6. The following pertaining arts are discovered and not used in this office action. Office reserves the right to use these arts in later actions.

a. Anderson; Robin L. et al. (US 7349956 B2) Systems and methods for integrating emulated and native code

b. Beckett; William et al. (US 7555421 B1) Supporting a SCSI device on a non-SCSI transport medium of a network

c. Carnahan, Jason et al. (US 20040230668 A1) Simultaneous sharing of storage drives on blade center

d. Gandhi; Amar S. et al. (US 7085814 B1) Systems and methods for capturing screen displays from a host computing system for display at a remote terminal

e. Johnson; Karl S. et al. (US 6249885 B1) Apparatus and method for detecting the reset of a node in a cluster computer system

Conclusion

7. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Muktesh G. Gupta whose telephone number is 571-270-

5011. The examiner can normally be reached on Monday-Friday, 8:00 a.m. -5:00 p.m., EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William C. Vaughn can be reached on 571-272-3922. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

MG

/William C. Vaughn, Jr./

Supervisory Patent Examiner, Art Unit 2444